

JavaScript Cheatsheet

Adding JavaScript to HTML

Inline Script (Avoid for Production)

```
<script>
  console.log("Hello World");
</script>
```

External JS File (Recommended)

```
<script src="app.js" defer></script>
```

- Use `defer` to ensure scripts run after HTML is parsed.
- Use `type="module"` for ES modules.

```
<script type="module" src="app.js"></script>
```

Variables

Modern way to declare variables:

```
let name = "Harry";    // Block-scoped, reassignable
const PI = 3.14;       // Block-scoped, cannot be reassigned
var oldWay = true;     // Function-scoped, avoid using
```

Functions

Function Declaration

```
function greet(name) {  
  return `Hello, ${name}!`;  
}
```

Arrow Functions (Modern)

```
const greet = (name) => `Hello, ${name}!`;
```

Default Parameters

```
function greet(name = "Guest") {  
  console.log(`Hello, ${name}`);  
}
```

DOM Manipulation

Selecting Elements

```
document.querySelector('#id');  
document.querySelectorAll('.class');
```

Changing Content

```
document.getElementById("elementID").textContent = "Hello World!";
```

Creating & Appending Elements

```
const div = document.createElement('div');
div.textContent = "New Element";
document.body.appendChild(div);
```

Console Output

```
console.log("Message");
console.error("Error message");
console.table([1, 2, 3]);
```

Conditional Statements

```
if (condition) {
  // code
} else if (otherCondition) {
  // code
} else {
  // code
}

switch(value) {
  case 'x':
    break;
  default:
}
}
```

Loops & Iteration

For Loop

```
for (let i = 0; i < 5; i++) console.log(i);
```

For...of (Modern)

```
for (const item of ['a', 'b', 'c']) console.log(item);
```

forEach

```
[1, 2, 3].forEach(num => console.log(num));
```

While & Do While

```
while (i < 5) i++;  
do { i++; } while (i < 5);
```

Strings

```
const str = "JavaScript";  
str.charAt(3);  
str.includes("Script");  
str.startsWith("Java");  
str.endsWith("pt");  
str.replace("Java", "Type");  
str.split("");  
str.toUpperCase();
```

Arrays

```
const arr = [1, 2, 3];
arr.push(4);
arr.pop();
arr.shift();
arr.unshift(0);
arr.includes(2);
arr.find(num => num > 1);
arr.filter(num => num > 1);
arr.map(num => num * 2);
arr.reduce((acc, cur) => acc + cur, 0);
```

Numbers & Math

```
Number.isNaN(value);
Math.ceil(4.2);
Math.floor(4.9);
Math.random(); // 0-1
Math.max(1, 5, 9);
Math.min(1, 5, 9);
```

Dates

```
const now = new Date();
now.getFullYear();
now.getMonth(); // 0-11
now.toISOString();
```

Events

```
document.querySelector('#btn').addEventListener('click', e => {  
  console.log('Button clicked', e);  
});
```

Common events: `click`, `input`, `change`, `submit`, `keydown`, `keyup`,
`mouseenter`, `mouseleave`

Error Handling

```
try {  
  throw new Error("Something went wrong");  
} catch (error) {  
  console.error(error);  
} finally {  
  console.log("Always runs");  
}
```

Async JavaScript

Promises

```
fetch('https://api.example.com')  
  .then(res => res.json())  
  .then(data => console.log(data))  
  .catch(err => console.error(err));
```

Async/Await

```
async function getData() {  
  try {  
    const res = await fetch('https://api.example.com');  
    const data = await res.json();  
    console.log(data);  
  } catch (err) {  
    console.error(err);  
  }  
}
```

Window Methods

```
alert("Hello");  
const confirmed = confirm("Are you sure?");  
const name = prompt("Enter your name");  
setTimeout(() => console.log("Timeout"), 1000);  
const interval = setInterval(() => console.log("Tick"), 1000);  
clearInterval(interval);
```

Modern ES6+ Features

Template Literals

```
const name = "Harry";  
console.log(`Hello, ${name}`);
```

Destructuring

```
const person = { name: "Harry", age: 25 };  
const { name, age } = person;
```

Spread & Rest Operators

```
const nums = [1, 2, 3];  
const copy = [...nums];  
function sum(...args) {  
  return args.reduce((a, b) => a + b);  
}
```

Modules

```
// export.js  
export const PI = 3.14;  
export default function greet() {  
  console.log("Hello");  
}  
  
// import.js  
import greet, { PI } from './export.js';
```

Debugging

```
debugger; // Pauses execution in DevTools
```